9. Introduction to Simile

Jerry Vanclay[1]

This module provides an introduction to Simile, a powerful modelling language with some innovative features. Simile is being developed by Dr Robert Muetzelfeldt, at the University of Edinburgh, with suggestions and testing by a number of users. The package is still under development, so it is getting better all the time, but it is already a sophisticated and stable package. Documentation is available at http://www.ierm.ed.ac.uk/simile, including some that was prepared when Simile was known by its previous name, AME (Agroforestry Modelling Environment). The latest version of the software can also be downloaded without charge from this Web site, for PCs with Windows 9x or computers running Linux. Dr Muetzelfeldt has long had an interest in efficient representation of ecological information, and his development of Simile was due in part because of his concern that with many models, the documentation, diagrams and computer implementation diverge. So he set out to build a modelling platform where the diagram was the model and the documentation, and he's close to achieving this goal. He's also provided the possibility to create a model that runs on a computer without the need for computer code or mathematical equations. However, there's no free lunch, and it is necessary to learn some of the standard notation used in systems dynamics to be able to use this package. This module provides introduces the systems dynamics notation, and then provides a worked example of developing a Simile model to represent a personal bank account. Some of the unique features of simile are then examined with reference to a simple forestry model. Finally, two other examples of forestry models are presented briefly.

1. SYSTEMS DYNAMICS NOTATION

Although an effort will be made to keep the jargon to a minimum, the user does need to learn some basic systems dynamics vocabulary to get started. In particular, a common understanding is needed of the following terms: compartment, flow, cloud, variable, influence and equation.

A *compartment*
- is represented by a rectangle
- represents amount of some substance (or occasionally non-substance things like height or position)
- should be labelled as object:substance, e.g. tree volume with units cu_m/ha (or no./ha, kgs, etc)
- unlike real compartments, can go negative, and has infinite capacity
- cannot receive an influence arrow – it changes only as a result of flows
- rate of change is the net effect of all inflows minus all outflows
- two connected compartments must have same substance, same units
- can only contain one substance.

A *flow*
- usually corresponds to a process, and must flow into or out of a compartment
- has units that correspond to compartment units per unit time
- can be negative (flow in reverse direction)
- may be one of several flows between two or more compartments, in either direction
- ideally represents a single process, so it is preferable to have one flow for each separately-analysable process.

[1] This module is a modification of lecture material prepared for students at Southern Cross University in March 2001.

A *cloud*
- is like a compartment (at start or end of a flow), but when its value is irrelevant, unknown, or unspecified
- corresponds to 'the outside world', and thus cannot receive or be the source of an influence arrow
- in Simile, clouds are created automatically when a flow starts nowhere or goes nowhere.

A *variable* can be
- a parameter that is 'constant' during a simulation (e.g. a coefficient in an equation)
- an intermediate variable, which can be both the source and recipient of influence arrows
- an output variable used only for reporting on model behaviour
- an exogenous variable (i.e. an 'external variable', just a function of time) that influences the model, but is not influenced by it (e.g. climatic factors).

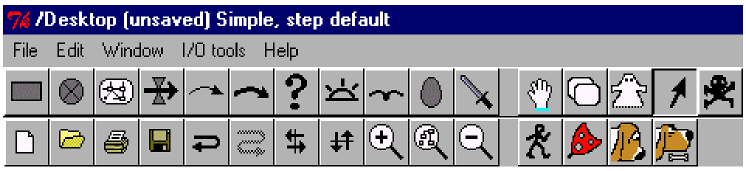
An *influence*
- usually corresponds to a link in an 'influence diagram'
- captures the idea that something affects something else
- formally, represents the fact that one term is used in the calculation of another
- can start from a compartment, flow or variable, and go to a flow or a variable (but *not* to a compartment).

Finally, it should be recognised that an *equation* may appear within a compartment, a flow or an influence. The equation
- says how a value for a variable is to be calculated
- often represents the relationship between one quantity (Y), and one or more influencing quantities (X_1, X_2, etc)
- uses standard algebraic expressions and conditional elements
- may include a sketched graph function or a tabulated function

In Simile, a single 'equation window' is used for all quantities (incl. parameters and initial compartment values).

2. GETTING STARTED

When the new version of Simile is opened, a window like the one opposite is obtained. One of the nice features of this version is that if you're not sure what a button does, hold the mouse over it, and an explanation will appear.

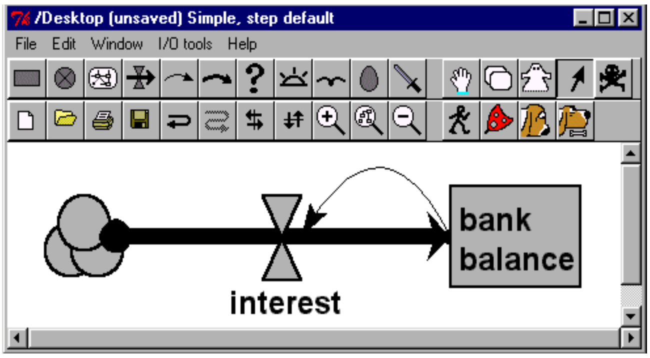


3. BUILDING A SIMPLE MODEL OF A BANK ACCOUNT

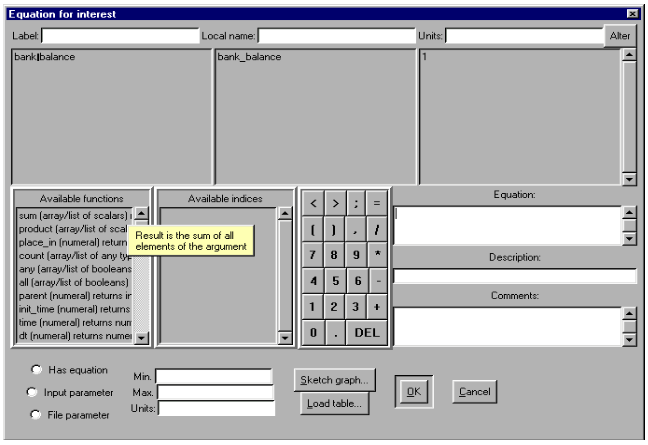
To build a simple simile model which will simulate changes in the level of savings over time in a bank account, follow the steps listed below:

1. Click on the COMPARTMENT tool (left-most rectangle) in the top toolbar.
2. Click in the centre of the 'Desktop' window to deposit a compartment symbol there. *Note that it is automatically labelled 'comp1'.*
3. Click on the FLOW tool (straight thick arrow) in the toolbar.
4. Drag a flow into the compartment symbol in the Desktop window.
   'Drag' means:move the mouse onto the Desktop window, depress the mouse button where the flow should begin, move the mouse to touch the compartment symbol (it should turn green), and release the mouse button. Note the flow is automatically labelled 'flow1'

5. Click on the INFLUENCE ARROW tool (curved thin arrow) in the toolbar.
6. Drag an influence arrow from the compartment symbol to the flow symbol in the Desktop window. *This indicates that the interest payments depend on (are influenced by) the bank balance.*
   Your diagram should now look like the one on the previous page.



7. Click on the POINTER tool (arrow inside a rectangle) in the toolbar.
8. Click on the compartment symbol in the Desktop window.
9. Use the Delete and/or backspace keys to remove the label ‘comp1’.
10. Type in a name for the compartment, e.g. balance.
11.Click on the flow symbol in the Desktop window.
12. Use the Delete and/or the backspace keys to remove the label ‘flow1’.
13. Type in a name for the flow, e.g. interest.
    We’ve finished defining the structure of the model, but have not initialized it; that’s why it is in red.
14. Double-click on the compartment symbol.
15. Enter the value 100 in the ‘Equation’ box.
    This states that the initial bank balance is $100.
16. Click on OK. *Notice the compartment is now black, indicating that it has been initialized.*
17. Double-click on the flow symbol (the double-triangle bit).
18. Enter the expression 0.1*bank_balance in the ‘Equation’ box.
    This expresses the fact that the annual interest is 10% of the balance. Notice that Simile offers help with functions if you place the mouse over them. Also note that you must type names exactly the same way as they appear in the equation box – it is often convenient to double-click on variable names rather than typing them.

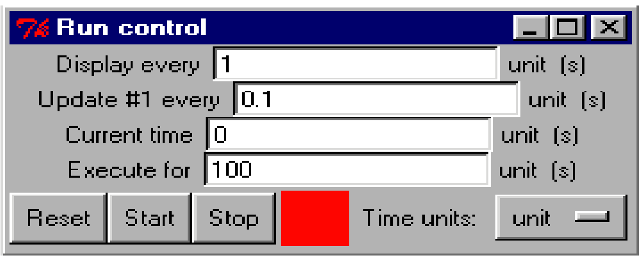


19. Click on OK.
    The model is now completely set up, and ready to run. It should look like this the window opposite.

4. RUNNING THE MODEL

This comprises the following steps:

1. Click on the word ‘File’ in the Desktop menu bar.
2. Select the ‘Run’ option in the File menu, then select the ‘In tcl’ sub-option. *Note that a run control dialogue window appears.*
3. Re-arrange the windows to avoid overlap (drag by top title bar).



4. Click in the ‘Update every’ edit field.
5. Delete the ‘0.1’ and replace it by ‘1’ (without the quotes).
6. Change the value 100 to 10 in the ‘Execute for’ edit field.
7. Open the ‘I/O tools’ menu in the Desktop window.

8. Select the 'Add tool' option, then the 'Plot value against time' sub-option.
   Note the little window that pops up telling you what to do now.
9. Click on the compartment in the Desktop window.
   This sequence specifies that you want to plot values for your bank balance against time.
10. Again, re-arrange the windows if necessary.
11. Click on 'Start' in the Run Control dialogue window.
    The model should now simulate according to the values in the run control window: 10 years, displaying the results every year, and calculating interest once per year. Observe the results in the window titled 'Current value for balance', and note that the actual value of balance is displayed in the box in the bottom-left corner.
12. Try re-sizing the graph window; scrolling up and down; and clicking on the 'Var' and 'Time' buttons.

5. CHANGING THE MODEL

This may be achieved in a number of ways.

By changing the equation

1. Click in the Desktop window to return to the model-design system.
2. Change the interest rate from 10% to 15%.
   You should realise that this involves selecting the pointer tool, double-clicking on the flow symbol in the Desktop window, and editing the interest equation, replacing 0.1 by 0.15.
3. Run the model again, and compare the results with the previous results.

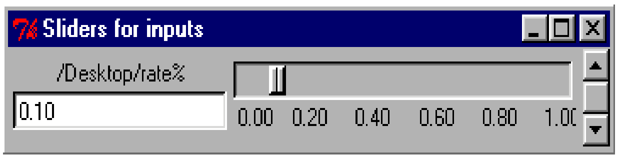
By changing a compartment value during a run

1. With the graph still on the screen, drag the slider near the y-axis down to 50. *This simulates the effect of removing some money from the bank account, leaving just $50*.
2. Click on 'Start' to continue the simulation.
   Note how the graph drops down the climbs back up again. Why do you think the line drops down at an angle rather than dropping down vertically?

With an explicit symbol on the model diagram

1. Click on the VARIABLE tool (circle with a cross inside it).
2. Click near the flow symbol in your model diagram to place a 'variable' symbol there.
3. Re-label this symbol as 'rate'.
4. Double-click on this symbol, enter the value 0.15 in the 'Equation' box, and click on OK.
5. Draw an influence arrow from the variable symbol to the flow symbol.
6. Replace the expression for interest (0.15*interest) by rate*balance
7. Run the model again.
   The results should be the same as before. You have now just changed the diagrammatic appearance of the model, not its underlying mathematical structure. In general, it's up to you whether you show model parameters explicitly or hide them in expressions

By adding a slider to the model



1. Click on the POINTER tool.
2. Double-click on the VARIABLE symbol 'rate' to open the equation window.
3. Click the button 'Input parameter' and supply the Min and Max.
4. Click on the 'Run' tool to re-run your model.
   Notice the appearance of a slider that enables you to vary the interest rate at any time during simulation.

By adding more variables to the model

As an exercise, add features to your model to represent regular deposits (e.g. your salary) and withdrawals (e.g. loan repayments) to your model.

6. SAVING THE MODEL

To save the model, take the following steps:

1. Click on the SAVE tool or select Save under File in the Desktop menu bar.
2. Find the required drive (normally A, C or your home drive) and select the required directory, as you would do in any Windows program.
3. Type the name for the file (e.g. model01). *Simile will automatically add the extension '.sml' to the file name*. *Older versions may add '.ame' or '.sim'.*
4. Click on 'Save'.

So far, the standard systems dynamics notation has been demonstrated. Now some more advanced features unique to Simile will be introduced.

7. SOME UNIQUE FEATURES OF SIMILE

Simile offers some special features that make it uniquely amenable for modelling and teaching. One of these is that the diagram is the model and the documentation, and offers a comprehensive and clear overview of a model, even to the uninitiated. Another unique aspect is the concept of multiple-instance submodels, including several special instances of submodels, both of which are now discussed.

A forestry example of diagram-is-model-is-documentation

Take a quick look at the following four illustrations of the same model implemented in different ways (BASIC, Excel, System dynamics, Simile). You don't need to understand all four approaches; these are presented to emphasise some of the strengths of Simile.

1) Denis Alder's size class model in Basic

```
SUB stproj(st())
'Updates a stand table of tree diameters by species using simple stand projection
'Get array bounds. Lower bound should be 1.
nsp% = UBOUND(st, 1)
ndc% = UBOUND(st, 2)
DIM st0(ndc%)
FOR j% = 1 TO nsp%
        'Keep old stand table as st0 to avoid overlap while updating
        FOR k% = 1 TO ndc%
            sto(k%) = st(j%, k%)
        NEXT k%
        'Get first class update using ingrowth function
        st(j%, k%) = ingrowth(j%) + st0(k%) * (1 – growth(j%, k%) – death(j%, k%))
        'Get updates to other classes
        FOR k% = 2 TO ndc%
            st(j%, k%) = st0(K%-1)*growth(j%, f%-1) + st0(k%)*(1 – growth(j%, k%) – death(k%, j%))
        NEXT k%
NEXT j%
END SUB
```

From Alder, D. (1995), *Growth Modelling for Mixed Tropical Forests*, p. 80.

2) The same model in Excel

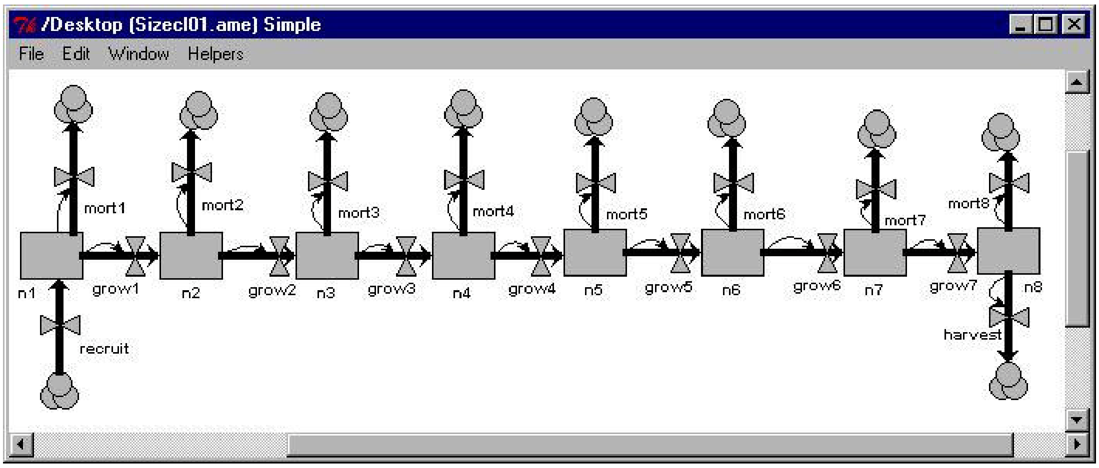
Diameter Class

From (cm)	5	10	20	40	60	80	100	120
To (cm)	=C3	=D3	=E3	=F3	=G3	=H3	=I3	+
Increment cm/yr	0.686	0.77	0.8	0.74	0.625	0.485	0.327	0.24
Outgrowth %/5yr	=5*B5/(B4-B3)	=5*C5/(C4-C3)	=5*D5/(D4-D3)	=5*E5/(E4-E3)	=5*F5/(F4-F3)	=5*G5/(G4-G3)	=5*H5/(H4-H3)	0
Mortality %/5yr	0.14	0.049	0.049	0.049	0.049	0.049	0.049	0.049
Harvest %								0.5
Year	←--- Trees/km^2 ---→							
0	500	321	232	146	164	136	64	14
=A12+5	=B12*(1-B$6-B$7)+87	=B12*B$6+C12* (1-C$6-C$7)	=C12*C$6+D12* (1-D$6-D$7)	=D12*D$6+E12* (1-E$6-E$7)	=E12*E$6+F12* (1-F$6-F$7)	=F12*F$6+G12* (1-G$6-G$7)	=G12*G$6+H12* (1-H$6-H$7)	=H12*H$6+I12* (1-I$6-I$7)*I$8
=A13+5	=B13*(1-B$6-B$7)+87	=B13*B$6+C13* (1-C$6-C$7)	=C13*C$6+D13* (1-D$6-D$7)	=D13*D$6+E13* (1-E$6-E$7)	=E13*E$6+F13* (1-F$6-F$7)	=F13*F$6+G13* (1-G$6-G$7)	=G13*G$6+H13* (1-H$6-H$7)	=H13*H$6+I13* (1-I$6-I$7)*I$8
=A14+5	=B14*(1-B$6-B$7)+87	=B14*B$6+C14* (1-C$6-C$7)	=C14*C$6+D14* (1-D$6-D$7)	=D14*D$6+E14* (1-E$6-E$7)	=E14*E$6+F14* (1-F$6-F$7)	=F14*F$6+G14* (1-G$6-G$7)	=G14*G$6+H14* (1-H$6-H$7)	=H14*H$6+I14* (1-I$6-I$7)*I$8

From Alder, D. (1995) *Growth Modelling for Mixed Tropical Forests*, p. 78.

3) The same model in a systems dynamic representation

This is an exact re-implementation of the spreadsheet model given above. It is implemented in Simile, but uses only standard system dynamic notation used in industry-standard packages like Stella (from Muetzelfeldt's submodel library).

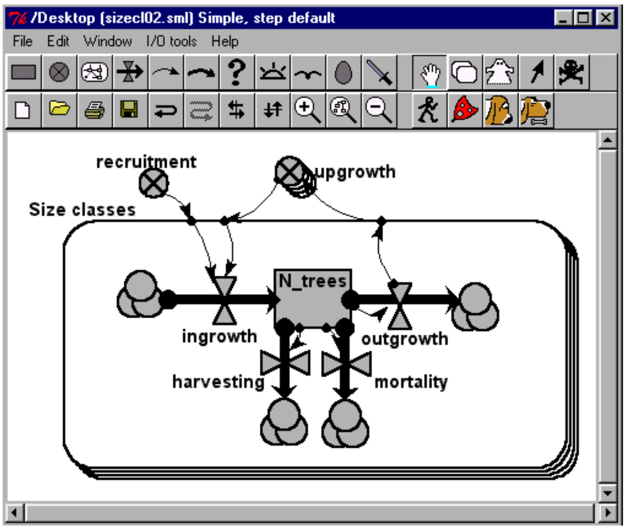


Initial values:
n1...n8 = 500, 321, 232, 146, 164, 136, 64, 14
Equations:
recruit = 87
mort1...mort8 = 0.14*n1, 0.049*n2...0.049*n8
grow1...grow8 = 0.686*n1, 0.385*n2, 0.2*n3, 0.185*n4, 0.156*n5, 0.121*n6, 0.082*n7, 0*n8
harvest = 0.5*n8

4) The same model in Simile, using a multiple-instance submodel

This is the same model, but using a multiple-instance submodel to represent a generic size-class. It has eight instances, since the published model has eight size-classes.

Initial values:
recruitment = 87
N_trees =
element([500,321,232,146,164,136,64,14]
,index(1))
Equations:
mortality = if index(1)==1 then
0.14*N_trees else 0.049*N_trees
outgrowth =
element([0.686,0.385,0.2,0.185,0.156,0.1
21,0.082,0],index(1))*N_trees
harvesting = if index(1)==8 then
0.5*N_trees else 0
upgrowth = [outgrowth]
ingrowth = if index(1)>1 then
element([upgrowth],index(1)-1)else 0



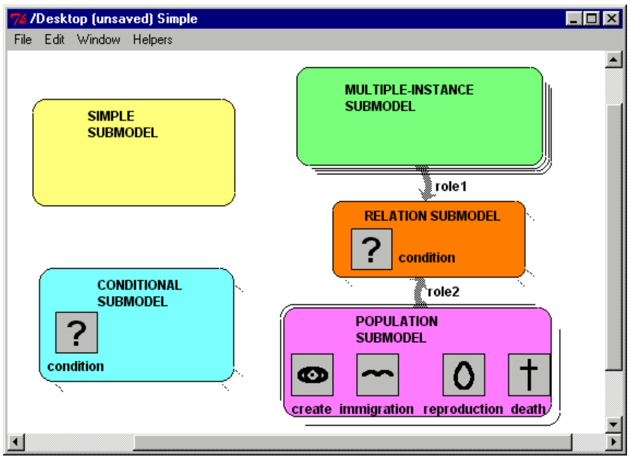
Which representation is clearer and easier to follow? Why?

Submodels

One of the powerful features you've just seen is the ability to create a submodel.

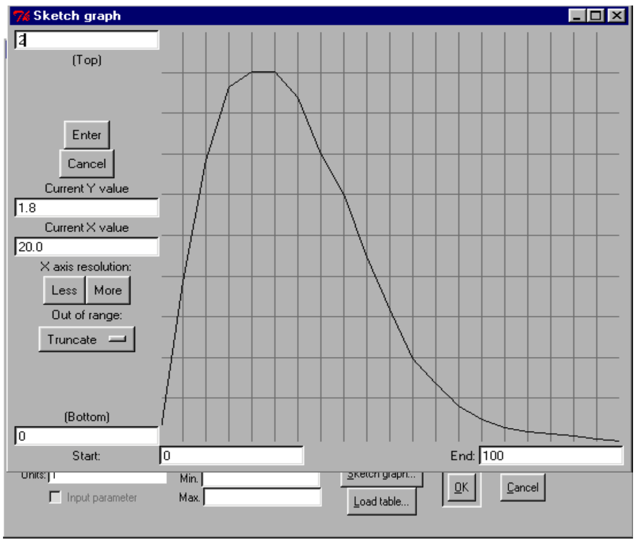
Submodels can be used to

- visually separate parts of the model
- control the appearance of a complex model
- enable separate saving and loading
- move parts of a model around
- create multiple instances
- specify relations between objects
- allow parts of the model to exist conditionally
- specify dynamically-varying populations.

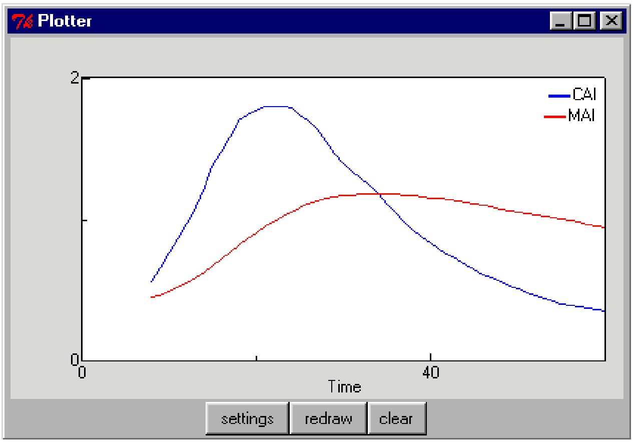


Using submodels to model individual trees

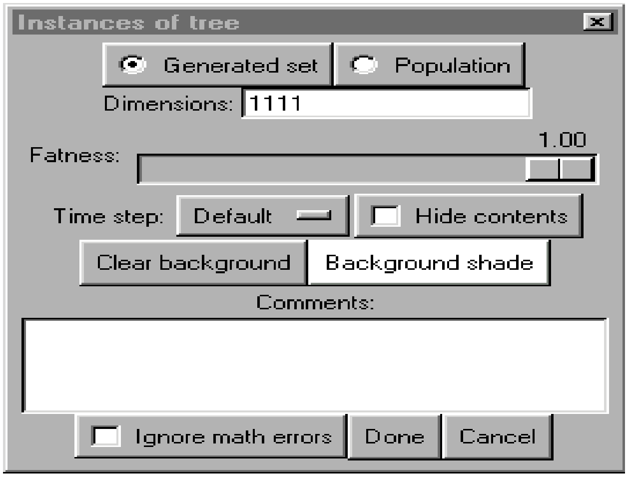
Let's build an individual tree growth model with Simile, starting with a simple model of the growth of a single tree. It's much like the bank-balance model illustrated in Section 3, but here the compartment is called diameter, and the flow is called grow. Suppose there is no established equation for tree growth, and that the model is to be constructed using a graph to represent growth. Double-click on the flow and click on the 'Sketch graph' button. The Sketch graph dialogue window appears, with its default straight line and 0,0 to 100,100 coordinates. Since growth is to be predicted from diameter (in



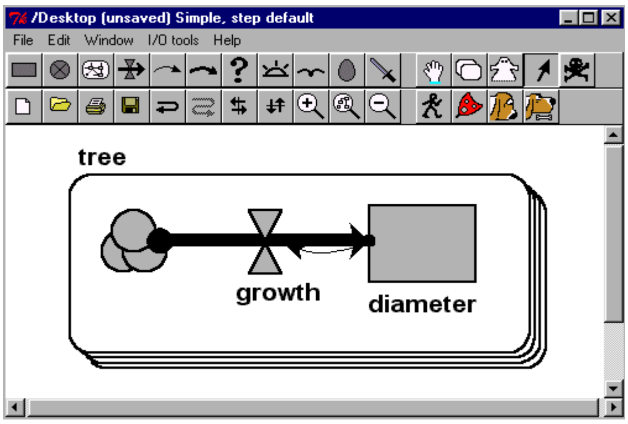
cm dbh), 0–100 is probably OK on the horizontal axis, but a more appropriate maximum is needed on the vertical axis (let's chose 2, a reasonable maximum when growth is expressed in cm/year). Next, 'draw' the response curve with the mouse, either by dragging, or by clicking on the grids. When this is finished, press Enter to return to the equation dialogue. Notice that "graph()" has appeared in the equation box; this must be completed by double-clicking on the local name diameter to complete the equation, or Simile won't know how the graph should be used.



Let's also compute the mean annual increment (MAI) and compare the current (CAI or growth) and MAI curves. Create a variable called MAI, and drag an influence arrow from diameter to MAI. Double-click on the variable MAI and enter the formula for MAI: "if time(1)==0 then 0 else diameter/time(1)". The function time(1) is a system function that returns the current simulated time. Notice that Simile uses the notation "==" to indicate a test of equality to discriminate it from the common usage of "=" indicating "takes the value of".



Remember how to run the model? (If not, see Section 4 above). An "I/O tool" is needed to illustrate the CAI and MAI curves. The screen opposite illustrates the plotter. To use this, select "I/O tools", ""Add tool" and "Plotter".



Now use the submodel tool to include all these details in a submodel. Suppose it is desired to model a plantation with 1111 trees (3 x 3 metre spacing). Click on the pointer tool and double click on the white space inside the model. A submodel dialogue box should appear, and you can specify the number of instances (1111), choose a colour if you wish, and add some comments. When this is done, the submodel diagram will have changed, now appearing like a pack of cards, to represent multiple instances.

If the screen does not appear like this, use the move tool and the zoomfit tool to tidy it up a bit.

Now try to compute the basal area. Use the variable tool to create a variable called BA outside the submodel, and drag an influence arrow from the diameter compartment to the BA variable. Double-click to open the equation dialogue. Notice that the influence arrow makes an *array* of diameters available to this variable. Double-click the *sum* function, double-click the local name [diameter], and type ^2 to square it. Next, move the cursor to the end and divide by 12732, which is equivalent to 40000/π. π is not yet available as a function in Simile. The figure 40000 converts diameter squared into radius squared, and cm^2 into m^2. Try running the model, and explore some of the I/O tools.

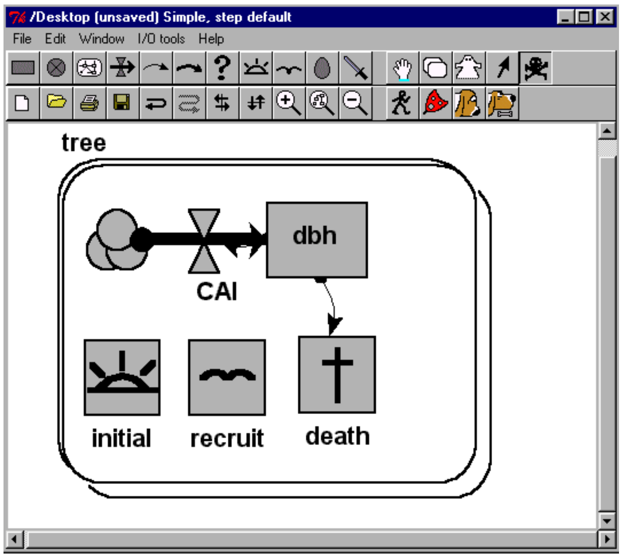
Let's adapt this model for uneven-aged forests, and regeneration. Double-click on the submodel again, and chose the Population button. When this is done, you'll notice that the 'pack of cards' looks 'slightly untidy', signifying that the number of instances can vary. A population submodel also activates several new symbols:

- *creation* tells Simile how many instances exist at the start of a simulation.

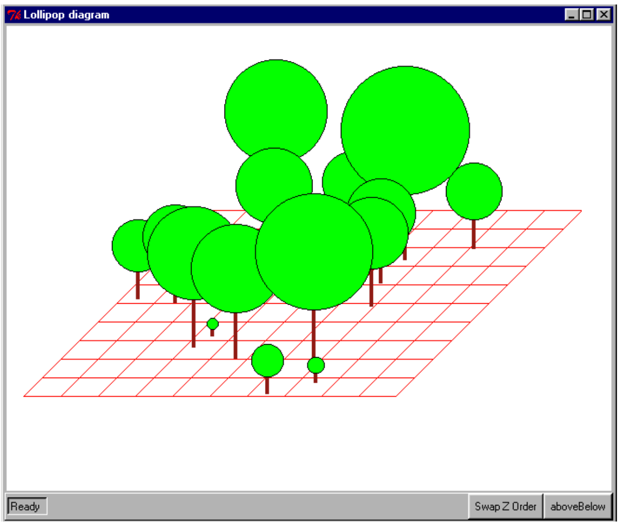
- *reproduction* creates 'offspring' for each existing instance; good for modelling regeneration in species with short-lived seed, and for modelling animal populations (the symbol is an egg).

- *immigration* creates new instances without regard to the existing instances; good for modelling regeneration from seed banks, of species with widely-dispersed seed, and animal migration.

- *loss* provides a way to model mortality, emigration, etc. (the symbol is a sword or a cross).



Alder's model will now be represented as a population model, so it can be compared with the illustrations above. A few numbers will need to be changed, to adopt an area of one hectare (not a square kilometre), and to use annual time steps instead of 5-year steps. Here's the model. Does it make sense to you? One thing that it portrays clearly is that growth and death depend on tree size (and tree size only), and everything else remains constant throughout the simulation.



One further embellishment will be made, namely adding x- and y- coordinates for each tree so that a 'lollipop' diagram can be created. Module 10 demonstrates how to use these Simile functions, developing a forestry example model. Also, some tutorials on Simile can be accessed on the Web site: http://www.ierm.ed.ac.uk/simile.

8. CONCLUDING COMMENTS

Simile is a powerful modeling and programming medium, which has proven valuable in modeling forestry systems. As with any computer programming language, or software package, there are a number of overheads involved in gaining familiarity with simile. The package is available as freeware, so anyone can obtain a copy and start using it. It is obviously necessary to learn the systems dynamics notation, and to become familiar with the menu items of the simile screen. After that, working through example applications such as the bank account model presented here provides initial confidence for the user. To become a competent user, it is essential to experiment with the Simile package, including the menu and sub-menus, and find out by trial and error what operations can be performed. Handy help facilities are available to assist this experimentation.